IN THE SPECIFICATION:

Replace paragraphs **[0006] – [0012], [0014], [0016], [0017],** and **[0019] – [0028]** and add paragraphs **[0016A], [0016B], [0027A], [0027B]** and **[0028A]** as follows:

[0006] The present invention implements a method used to secure computer files on a ~~file server~~ computer using ~~dual-key~~ encryption ~~technologies without requiring the exchange of encryption keys with external users~~. The method may be embedded within one or more computer-readable programs written in a programming language, such as Perl, and running on a ~~web server~~ computer. The method may employ the use of a single encryption/decryption key pair that is stored on the same ~~web server~~ computer to encrypt files received from ~~external~~ users on an inbound path and to decrypt files delivered to ~~external~~ users on an outbound path. All ~~inbound and output~~ encryption and decryption occurs in real time in a memory subsystem of the ~~web server, which may include Random Access Memory (RAM)~~ computer. As a result, no unencrypted version of an electronic file needs to be created within the computer using the present invention. The method and system do not require the use of any specific ~~dual-key or public-private key~~ encryption product, operating system or environment.

[0007] In a preferred embodiment of the present invention, a method of ~~encrypting and decrypting~~ transferring an electronic file on a ~~web-based~~ computer system includes receiving, by a computer system, an electronic data file, where the computer system includes a memory subsystem and a plurality of memory locations, encrypting the data file in the memory subsystem, and storing the encrypted data file in one or more ~~of the plurality of~~ memory locations~~, retrieving the encrypted data file from the one or more memory locations, decrypting the encrypted data file in the memory subsystem, and displaying the decrypted data file on a web browser~~. Encrypting the data file occurs without assistance from a user and without requiring

knowledge of the encryption algorithm by the user. ~~In an embodiment, the memory subsystem includes random access memory.~~ In an embodiment, the receiving step is performed using a SSL/HTTPS protocol. ~~In an embodiment, the transmitting step is performed using a SSL/HTTPS protocol.~~ In an embodiment, the method may further include receiving a username and a password from a user device and verifying that the username and password correspond to a pre-defined user having access to the computer system and/or an encrypted file. In an alternate embodiment, the method further includes retrieving the encrypted data file from the one or more memory locations, decrypting the file, modifying the decrypted data file, encrypting the modified file, and storing the modified data file in the one or more memory locations.

[0008] A method of transferring an electronic file on a computer system includes retrieving, from a computer system including a memory subsystem and a plurality of memory locations, an encrypted data file from one or more memory locations, decrypting the encrypted data file in the memory subsystem, and displaying the decrypted data file. Decrypting the encrypted data file occurs without assistance from a user and without requiring knowledge of the decryption algorithm by the user. In an embodiment, the transmitting step is performed using a SSL/HTTPS protocol. In an embodiment, the method may further include~~, prior to the receiving step~~ receiving a ~~username~~ user name and a password from ~~an external~~ a user device and verifying that the ~~username~~ user name and password correspond to a pre-defined user having access to the computer system and/or the encrypted file. In an alternate embodiment, the method further includes~~, between the storing step and the retrieving step~~ retrieving the encrypted data file from the one or more memory locations, ~~analyzing~~ decrypting the ~~encrypted data~~ file, modifying the ~~analyzed~~ decrypted data file, encrypting the modified data file and storing the modified data file in the one or more memory locations.

[0009] In an alternate embodiment, a method of ~~encrypting and decrypting~~ transferring an electronic data file on a ~~web-based~~ computer system includes receiving, by a ~~web server~~ first computer, an electronic data file, where the ~~web server~~ first computer includes a memory subsystem, encrypting the data file in the memory subsystem, transmitting the encrypted data file to a ~~file server~~ second computer having a plurality of memory locations, and storing the encrypted data file in one or more ~~of the plurality of~~ memory locations~~, retrieving the encrypted data file from the one or more memory locations, transmitting the encrypted data file to the web server, decrypting the encrypted data file in the memory subsystem, and displaying the decrypted data file on a web browser~~. Encrypting the data file occurs without assistance from a user and without requiring knowledge of the encryption algorithm by the user. In an embodiment, the method further includes retrieving the encrypted data file from the one or more memory locations, decrypting the data file, modifying the decrypted data file, encrypting the modified data file, and storing the modified data file in the one or more memory locations. In an alternate embodiment, the method further includes~~, between the storing step and the retrieving step,~~ retrieving the encrypted data file from the one or more memory locations, transmitting the encrypted data file to a third computer ~~back-end data processing server~~, ~~analyzing, by the back-end data processing server, the encrypted data file,~~ decrypting the data file, modifying~~, by the back-end data processing server,~~ the ~~analyzed~~ data file, encrypting the modified data file, transmitting the encrypted modified data file to the second computer ~~file server~~, and storing the encrypted modified data file in the one or more memory locations.

[0010] In ~~a preferred~~ an embodiment, a ~~system for encrypting and decrypting~~ method for transferring an electronic data file on a computer system includes retrieving, from a ~~web server for encrypting a data file and decrypting~~ first computer having a plurality of memory

locations, an encrypted data file from one or more memory locations, ~~the web server~~ transmitting the encrypted data file to a second computer having a memory subsystem, ~~a file server, electrically connected to the web server, for storing the encrypted data file, the file server having a plurality of memory locations, and a back-end data processing server, electrically connected to the file server, for modifying the encrypted data file~~ decrypting the encrypted data file in the memory subsystem, and providing access to the decrypted data file. ~~The web server includes a computer process for receiving the data file from an external user device, encrypting the data file in the memory subsystem, and transmitting the encrypted data file to a file server. The file server includes a computer process for receiving the encrypted data file from the web server, storing the encrypted data file in one or more of a plurality of memory locations, retrieving the encrypted data file from the one or more memory locations, and transmitting the encrypted data file to the back-end data processing server. The back-end data processing server includes a computer process for receiving the encrypted data file from the file server, analyzing the encrypted data file, modifying the analyzed data file, and transmitting the modified data file to the file server. In a further embodiment, the computer process of the file server further includes receiving the modified data file from the back-end data processing server, storing the modified data file in the one or more memory locations, retrieving the modified data file from the one or more memory locations, and transmitting the modified data file to the web server. In a further embodiment, the computer process of the web server further includes receiving the modified data file from the file server, decrypting the modified data file in the memory subsystem, and displaying the decrypted data file on a web browser.~~ Decrypting the data file occurs without assistance from a user and without requiring knowledge of the decryption algorithm by the user. In an alternate embodiment, the method further includes retrieving the encrypted data file from

the one or more memory locations on the first computer, transmitting the encrypted data file to a third computer, decrypting the data file, providing access to or modifying the decrypted data file, encrypting the modified data file, transmitting the encrypted modified data file to the first computer, and storing the modified data file in the one or more memory locations.

[0011] ~~In an alternate embodiment, a system for encrypting and decrypting~~ In a preferred embodiment, a system for transferring an electronic data file includes a ~~web server for encrypting a data file and decrypting~~ first computer configured to encrypt a data file and decrypt an encrypted data file, the ~~web server~~ first computer having a memory subsystem, ~~and a file server~~ and a second computer, electrically connected to the ~~web server, for storing~~ first computer, and configured to store the encrypted data file, the ~~file server~~ second computer having a plurality of memory locations. The ~~web server includes a computer process for receiving the data file from an external~~ first computer comprises a processor configured to receive the data file from a user device, encrypt~~ing~~ the data file in the memory subsystem, and transmit~~ting~~ the encrypted data ~~file to the file server. The file server includes a computer process for receiving the encrypted data file from the web server, storing the~~ file to the second computer. The second computer comprises a processor configured to receive the encrypted data file from the first computer and store the encrypted data file in one or more ~~of the plurality of memory locations, retrieving the encrypted data file from the~~ memory locations. Encrypting the data file occurs without assistance from a user and without requiring knowledge of the encryption algorithm by the user. In an embodiment, the system further includes a third computer, electrically connected to the first computer, and configured to modify the encrypted data file. In the embodiment, the processor of the second computer is further configured to retrieve the encrypted data file from the one or more memory locations, ~~and transmitting the encrypted data file to the web server. In~~

~~a further embodiment, the computer process of the web server further includes receiving the encrypted data file from the file server, decrypting the encrypted data file in the memory subsystem, and displaying the decrypted data file on a web browser. In an alternate embodiment, the computer process of the file server further includes, between the storing step and the retrieving step, retrieving the encrypted~~ transmit the encrypted data file to the third computer, receive an encrypted modified data file from the third computer, and store the modified data file in the one or more memory locations. In the embodiment, the third computer comprises a processor configured to receive the encrypted data file from the second computer, decrypt the data file, modify the decrypted data file, encrypt the modified data file, and transmit the modified data file to the second computer. In an alternate embodiment, the processor of the second computer is further configured to retrieve the modified data file from the one or more memory locations, ~~analyzing the encrypted data file, modifying the analyzed data file, and storing the modified data file in the one or more memory locations.~~ and transmit the encrypted modified data file to the first computer. In an alternate embodiment, the first computer is further configured to receive the encrypted modified data file from the second computer, decrypt the modified data file in the memory subsystem, and display or provide access to the decrypted data file. Decrypting the encrypted data file occurs without assistance from a user and without requiring knowledge of the decryption algorithm by the user.

[0012] In an alternate embodiment, a system ~~of for encrypting and decrypting an electronic data file includes a server including a memory subsystem, a plurality of memory locations, and a computer process for receiving a data file from an external user device, encrypting the data file in a memory subsystem, storing the encrypted data file in one or more of a plurality of memory locations, retrieving the encrypted data file from the one or more memory~~

~~locations, decrypting the encrypted data file in the memory subsystem, and displaying the decrypted data file on a web browser. In a further embodiment, the computer process further includes, between the storing step and the retrieving step, retrieving the encrypted data file from the one or more memory locations, analyzing the encrypted data file, modifying the analyzed data file, and storing the modified data file in the one or more memory locations.~~ for transferring an electronic file comprises a receiving system for receiving an electronic date file, an encryption system for encrypting the electronic date file, a decryption system for decrypting the electronic data file, a memory system for storing the encrypted data file, and transmitting system for transmitting the decrypted data file, wherein the system is configured to encrypt and decrypt the electronic date file without assistance from the user or knowledge of the decryption algorithm used. In this embodiment, the systems may be deployed on one or more computers, servers or networks.

[0014] The invention may take form in various components and arrangements of components, and in various steps and arrangements of steps. The drawings are only for purposes of illustrating the preferred embodiments and are not to be construed as limiting the invention. In the drawings, common elements are identified by common reference numerals.

[0016] FIG. 2 illustrates a data flow diagram of the inbound flow of files sent from an external user ~~computer~~ and the outbound flow of files to an external user ~~computer~~ according to an embodiment of the present invention implemented on a three-tiered architecture.

[0016A] FIG. 3 illustrates a data flow diagram of the inbound flow of files sent from an external user and the outbound flow of files sent to an external user according to an embodiment of the present invention implemented on a two-tiered architecture.

[0016B] FIG. 4 illustrates a data flow diagram of the inbound flow of files sent from an external user and the outbound flow of files sent to an external user according to an embodiment of the present invention implemented on a single computer.

[0017] FIG. ~~3~~ 5 shows a program logic diagram for two computer program applications according to an embodiment of the present invention.

[0019] It must also be noted that as used herein and in the appended claims, the singular forms "a", "an", and "the" include plural reference unless the context clearly dictates otherwise. Thus, for example, reference to a "computer" or "server" is a reference to one or more computers or servers and equivalents thereof known to those skilled in the art, and so forth. Unless defined otherwise, all technical and scientific terms used herein have the same meanings as commonly understood by one of ordinary skill in the art. Although any methods similar or equivalent to those described herein can be used in the practice or testing of embodiments of the present invention, the preferred methods are now described. All publications mentioned herein are incorporated by reference. Nothing herein is to be construed as an admission that the invention is not entitled to antedate such disclosure by virtue of prior invention.

[0020] FIG. 1 depicts an exemplary diagram of the computer architecture and network connections used to implement an embodiment of the present invention. A user computer **101** may be connected to a computer network **102**. The computer network **102** may include, without limitation, the Internet, an intranet, or any other interconnected network of

computers. The connection of the user computer **101** to the computer network **102** may be achieved by any ~~standard~~ communication means **103** including, but not limited to, a dialup service, a cable connection, a digital subscriber line, an Ethernet network interface, an Asynchronous Transfer Mode network interface, a wireless service, or similar technologies. ~~A web server 3 running~~ The communication means may utilize any communication protocol, although a secure protocol such as HTTPS is preferred. A first server **104**, which may run, for example, a standard ~~http/https~~ HTTP/HTTPS web server application, may be used to transmit information that may be displayed on web pages to the user computer **101**. A ~~file~~ second server ~~4~~ **105** may store a plurality of ~~incoming files until they are retrieved by a back-end data processing server 5. In addition,~~ files which may be retrievable by a third server **106** or the ~~file~~ first server **104**. ~~may store outgoing files until they are retrieved by the user and sent to the user computer 1. A back-end data processing~~ The third server ~~5~~ **106** may be used to host special purpose applications that may transform or modify encrypted files and generate outgoing user-deliverable files. Optimally, the system may be configured with one or more firewalls, such as firewalls **107** and **108**, to control the flow of data between components of the system. It should be apparent to one skilled in the art that the servers identified in FIG. 1 may be physical servers such as individual computers or logical servers.

[0021] The computer/network architecture depicted in FIG. 1 is only one of many configurations that may be used to implement the method and system of the present invention. For example, the method or system may be implemented using only two ~~servers~~ computers, such as a ~~web server and a combined file and back-end data processing server~~ first computer performing the functions of the first server **104** and a second computer performing the functions of the second server **105** and the third server **106**. An example of such an embodiment is

depicted in FIG. 3. Moreover, the method or system may be implemented entirely within a single ~~server, such as a web server~~ computer that performs the functions of all three ~~function~~ servers described in reference to FIG. 1~~: web server, file server and back-end data processing server~~. An example of such an embodiment is depicted in FIG. 4. However, the network architecture depicted in FIG. 1, and described in reference thereto, is preferred because it maximizes security by separating the data flow and processing across machines that may be separated by firewalls.

[0022] FIG. 2 illustrates a data flow diagram of the inbound flow of files sent from ~~an external~~ a user computer **101** and the outbound flow of files to ~~an external~~ a user computer **101** according to an embodiment of the present invention. The user computer **101** ~~may access a login page~~ may access a first server **104** via communication means **103** to a network **102** connected to the first server **104**. The user computer **101** may access the first server **104** by way of a login page **206** of a service provider's website using any web browsing application, such as Netscape Navigator or Microsoft's Internet Explorer. ~~A~~ For example, a user may supply an assigned username and password when accessing the login page **206** in order to access the ~~web~~ first server ~~3~~ **104**. The transmission of the login page **206** and all subsequently described pages and files transmitted between the user computer **101** and the ~~web~~ first server ~~3 may~~ **104** will utilize ~~the~~ a secure ~~SSL/HTTPS~~ protocol ~~standard~~ such as SSL/HTTPS. The login page **206** of the preferred embodiment may be used to provide an additional layer of security. However, the login page **206** may be removed where user authentication via the submission of a username and/or a password is unnecessary, but encryption/decryption-on-demand is still required.

[0023] A user may select a file stored locally on the user computer **101** and submit the file for processing by the ~~web~~ first server ~~3~~ **104** via a file upload web page **207**. The

file upload process may be achieved through use of a standard ~~HTML tag, such as <form><input type="file" name="filename"></form>~~ techniques including use of a mark language tag, such as an HTML tag. The upload transmission may be securely transmitted via use of ~~the~~ a secure protocol, such as SSL/HTTPS ~~protocol standard~~, which provides an additional layer of security to the transmission environment. In an alternate embodiment, the SSL/HTTPS standard is not used for the transmission of one or more of the transmitted files between the user computer **101** and the ~~web~~ first server ~~3~~ 104.

[0024] A computer program **208** written in a computer-recognizable language, such as Perl, and stored on the ~~web~~ first server ~~3~~ **104**, may be used to process an incoming electronic data file. The process of encrypting the program is depicted in FIG. ~~3~~ 5a. The electronic data file may be processed by ~~reading~~ placing the data file in unencrypted form ~~16~~ **501** ~~from a buffer on the web server 3~~ into a memory subsystem of the ~~web~~ first server **104**. The memory subsystem may include one or more memory devices, including, without limitation, Random Access Memory (RAM), Dynamic Random Access Memory (DRAM), RDRAM, SDRAM, a CPU embedded cache, or any equivalents thereof. The content of the data file may then be encrypted ~~17~~ **502** in the memory subsystem via a system call to an encryption application, such as PGP. The encrypted data content ~~may be~~ is then saved ~~18~~ **503** to a file on the ~~web~~ first server ~~3~~ **104**. ~~The~~ Returning to FIG. 2, the encrypted data file may then be transferred **209** from the ~~web~~ first server ~~3~~ **104** to the ~~file~~ second server **4** **105**. This transfer may be performed via a File Transfer Protocol (FTP) program or any similar program for transferring files between servers.

[0025] ~~In an alternate embodiment, a computer application environment other than Perl may be used to implement the present invention. In fact, any application environment~~

~~permitting direct system calls (e.g., to an encryption utility) and Common Gateway Interface (CGI) interactions with a web server may be used. Moreover, the present invention may be implemented via the use of dual-key encryption technologies other than PGP or through the use of single-key or other encryption methodologies.~~

[0026] Once the encrypted data file is stored on the ~~file~~ second server **4** **105**, additional processing of the encrypted data file on the ~~back-end data processing~~ third server ~~5~~ **106** may be performed. Such additional processing is optional to the present invention. The additional processing may include ~~using a FTP program to send~~ transmitting **210** the encrypted data file as the input file 220 for the additional processing, from the ~~file~~ second server **4** **105** to the ~~back-end data processing~~ third server ~~5~~ **106**. The encrypted data file is then decrypted on the third server 106. The decrypted data file may then be accessed, analyzed, modified and/or rewritten ~~11 by the back-end data processing server 5, and transferred back 12 to the file server 4~~ **211** by the third server **106**. When processing on the third server **106** is completed, the data file is then encrypted, and transmitted **212** as the output file of the additional processing **225** to the second server **105** as an encrypted user-deliverable data file.

[0027] When requested by a user, the encrypted ~~user-deliverable~~ data file may be ~~transferred~~ transmitted **213** from the ~~file~~ second server **4** **105** to the ~~web~~ first server ~~3~~ **104**. This may be accomplished by using a FTP program. A computer program **214** written in a computer-recognizable language, such as Perl, and stored on the ~~web~~ first server ~~3~~ **104**, may be used to decrypt the outgoing encrypted user-deliverable data file. The process of decrypting the file is depicted in FIG. ~~3~~ 5a. The encrypted user-deliverable data file may be read, in encrypted form ~~19 from a buffer on the web server 3~~ **510** into the memory subsystem. The file content may be decrypted in the memory subsystem via a system call to a decryption application, such as PGP,

and the encrypted data file ~~may be~~ deleted from the system ~~20~~ **511**. The decrypted content in the memory subsystem ~~may~~ is then available to be downloaded ~~21~~ to the user~~'s browser 15 via a buffer on the web server 3.~~ via a secure protocol such as HTTPS **512**. Returning to FIG. 2, once the data file has been decrypted it is then available to be downloaded via a file download page **215**.

[0027A] In an alternate embodiment, the system of the present invention may be implemented on a two-server architecture. As shown in FIG. 3, the user computer **101** connects to a first server **104** via a network **102**. Access to the first server **104** is provided via the user log-in page **206**. The user may then submit a data file via the file upload page **207**. The computer program **208** encrypts the data file and the encrypted data file is transmitted **209**, such as via an FTP command, to a second server **305**. The encrypted data file is received by the second server **305** and stored in one or more memory location on the second server **305**. Optional data processing may then be performed where such processing occurs via the second server **305**. If such optional processing is performed, the encrypted data file is decrypted **310**, the optional processing is performed **311** and the data file is then encrypted **312** again. When the user then requests access to the data file, the data file is retrieved **213** from the second server **305**, such as via an FTP command. The computer program **214** decrypts the data file which is then available to be downloaded via a file download page **215**. The decrypted file is then transmitted to the user computer **101**, via the network **102** utilizing a secure protocol such as HTTPS.

[0027B] Although the computer programs **208** and **214** have been described in embodiments in which the program has been implemented in the Perl programming language, any programming language or application environment permitting direct system calls (e.g., to an

encryption utility) and Common Gateway Interface (CGI) interactions with a computer may be used. Moreover, although the present invention has been described with the use of PGP encryption, other encryption technologies will be equally effective including dual-key encryption, single-key encryption, hardware key encryption, or other encryption methodologies.

[0028] ~~The two~~ Furthermore, the computer programs ~~8,~~ **208** and **214** may perform additional functions that are not essential to the implementation of the present invention. The additional, non-essential functions are part of the preferred embodiment of the present invention, however, and are referenced herein to show the implementation of the preferred embodiment. The additional, non-essential functions ~~in computer program 8~~ may include, without limitation, ~~the~~ a user authentication process including the reception of a username and password~~. The additional, non-essential functions in computer program 14 may include, without limitation, a~~; means for creating a web page (dynamically) listing all available user-deliverable files and allowing the user to choose which file to decrypt and download; data compression; or other applications to efficiently or effectively receive, store, transmit or otherwise manage electronic data files.

[0028A] In an alternate embodiment, the present invention may be implemented on a single server. As shown in FIG. 4, a user may access the system via a user computer **101** connected to a server computer **401** via a network **102**. The user may access the server computer **401** via a user login screen **206** and upload to the server computer **401** a data file via a file upload page **207**. The data file will be encrypted by a computer program **208** and stored in one or more memory locations **402** on the server computer **401**. Optionally, additional processing may be performed on the data file while the file resides on the server computer **401** by decrypting the data file **403**, performing such optional processing **404**, and encrypting the data file **405** again.

When the user requests access to the data file, the computer program **214** decrypts the data file and the data file is available to be downloaded via a file download page **215**. The decrypted file is then transmitted to the user computer **101** via the network **102** utilizing a secure protocol such as HTTPS.